

/* **Program7:** Design, Develop and Implement a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: USN, Name, Branch, Sem, PhNo

- a. Create a SLL of N Students Data by using front insertion.
- b. Display the status of SLL and count the number of nodes in it
- c. Perform Insertion / Deletion at End of SLL
- d. Perform Insertion / Deletion at Front of SLL(Demonstration of stack)*/

```
#include <stdio.h>
#include<string.h>
#include<stdlib.h>
struct student
{
    char usn[12];
    char name[25];
    char branch[25];
    int sem;
    int phone_no;
    struct student *link;
};
typedef struct student * STUD;

STUD read_data()
{
    char usn[12],name[25],branch[25];
    int sem,phone_no;
    STUD temp;
    temp=(STUD)malloc(sizeof(struct student));
    printf("Enter the Students Details:\n");
    printf("Enter USN\n");
    scanf("%s",usn);
    strcpy(temp->usn,usn);
    printf("Enter Name\n");
    scanf("%s",name);
    strcpy(temp->name,name);
    printf("Enter Branch \n");
    scanf("%s",branch);
    strcpy(temp->branch,branch);
    printf("Enter Semester\n");
    scanf("%d",&sem);
    temp->sem=sem;
    printf("Enter Phone Number\n");
    scanf("%d",&phone_no);
    temp->phone_no=phone_no;
```

```
temp->link=NULL;
return temp;
}
```

```
STUD insert_front(STUD first)
{
    STUD temp;
    temp=read_data();
    temp->link=first;
    return temp;
}
```

```
STUD insert_end(STUD first)
{
    STUD temp,prev;
    temp=read_data();
    if(first==NULL)
        return temp;
    prev=first;
    while(prev->link!=NULL)
        prev=prev->link;
    prev->link=temp;
    return first;
}
```

```
STUD delete_front(STUD first)
{
    STUD cur;
    if(first==NULL)
    {
        printf("List is empty\n");
        return first;
    }
    cur=first;
    first=first->link;
    free(cur);
    return first;
}
```

```
STUD delete_end(STUD first)
{
    STUD prev,cur;
    if(first==NULL)
    {
```

```

    printf("List is empty\n");
    return first;
}
prev=NULL;
cur=first;
while(cur->link!=NULL)
{
    prev=cur;
    cur=cur->link;
}
prev->link=NULL;
free(cur);
return first;
}

```

```

void display(STUD first)
{
    STUD temp;
    int count=0;
    if(first==NULL)
    {
        printf("List is empty\n");
        return;
    }
    printf("USN\tNAME\tBRANCH\tSEM\tPHONE NO.\n");
    temp=first;
    while (temp!=NULL)
    {
        printf("%s\t%s\t%s\t%d\t%d\n",temp->usn,temp->name,temp->branch,temp->sem,temp->phone_no);
        temp=temp->link;
        count++;
    }
    printf("The number of nodes in SLL=%d\n",count);
}

```

```

int main()
{
    int ch,i,n;
    STUD first=NULL;
    //clrscr();
    printf("Creation of SLL of N Students\n");
    printf("Enter the number of students\n");
    scanf("%d",&n);
    for(i=1;i<=n;i++)

```

```

        first=insert_front(first);
        printf("SLL Created Successfully!!!\n");
        display(first);
while(1)
{
    printf("1.Display\n 2.Insert End\n 3:Delete End\n 4:Insert Front\n 5:Delete Front\n 6:Exit\n");
    printf("Enter the choice\n");
    scanf("%d",&ch);
    switch(ch)
    {
        case 1: display(first);
        break;
        case 2: first=insert_end(first);
        printf("Node Inserted at the End\n");
        break;
        case 3: first=delete_end(first);
        printf("Node deleted at the End\n");
        break;
        case 4: first=insert_front(first);
        printf("Node Inserted at Front\n");
        break;
        case 5: first=delete_front(first);
        printf("Node deleted at Front\n");
        break;
        case 6: exit(0);
        break;
        default:
        printf("INVALID CHOICE !");
    }
}
}

```